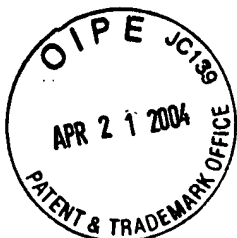


00862.023492



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	:	Examiner: Not Yet Assigned
TOSHIHIRO KOBAYASHI ET AL.	)	
	:	Group Art Unit: NYA
Application No.: 10/786,116	)	
	:	
Filed: February 26, 2004	)	
	:	
For: INFORMATION PROCESSING	)	
METHOD AND APPARATUS	:	April 20, 2004

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Sir:

In support of Applicants' claim for priority under 35 U.S.C. § 119, enclosed is  
a certified copy of the following Japanese application:

2003-054140, filed February 28, 2003.

Applicants' undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

Respectfully submitted,



Attorney for Applicants

LOCK SEE YU-JAHNES

Registration No. 38,667

FITZPATRICK, CELLA, HARPER & SCINTO  
30 Rockefeller Plaza  
New York, New York 10112-3800  
Facsimile: (212) 218-2200

NY\_MAIN 422750v1

日本国特許庁  
JAPAN PATENT OFFICE

10/786,116

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2003年 2月28日

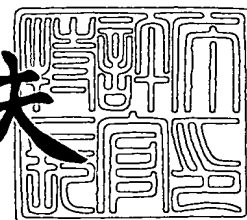
出願番号  
Application Number: 特願2003-054140  
[ST. 10/C]: [JP 2003-054140]

出願人  
Applicant(s): キヤノン株式会社

2004年 3月15日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



【書類名】 特許願

【整理番号】 253471

【提出日】 平成15年 2月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/00

【発明の名称】 情報処理方法及び装置

【請求項の数】 13

【発明者】

    【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会  
社内

    【氏名】 小林 俊広

【発明者】

    【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会  
社内

    【氏名】 藤木 真和

【発明者】

    【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号 キヤノン株式会  
社内

    【氏名】 大島 登志一

【特許出願人】

    【識別番号】 000001007

    【氏名又は名称】 キヤノン株式会社

【代理人】

    【識別番号】 100076428

    【弁理士】

    【氏名又は名称】 大塚 康德

    【電話番号】 03-5276-3241

## 【選任した代理人】

【識別番号】 100112508

## 【弁理士】

【氏名又は名称】 高柳 司郎

【電話番号】 03-5276-3241

## 【選任した代理人】

【識別番号】 100115071

## 【弁理士】

【氏名又は名称】 大塚 康弘

【電話番号】 03-5276-3241

## 【選任した代理人】

【識別番号】 100116894

## 【弁理士】

【氏名又は名称】 木村 秀二

【電話番号】 03-5276-3241

## 【手数料の表示】

【予納台帳番号】 003458

【納付金額】 21,000円

## 【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0102485

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理方法及び装置

【特許請求の範囲】

【請求項 1】 情報伝達媒体を介して通信可能な複数のプロセスによって、複数のデータアイテムで構成されたデータを共有するシステムにおいて、特定プロセスによるデータアイテムの排他的制御権を設定するための情報処理方法であって、

排他的制御権を設定するべく所望のデータアイテムを指示する第 1 指示工程と

、  
前記複数のデータアイテムの階層構造情報に基づいて、前記第 1 指示工程で指示されたデータアイテムよりも下位の階層に存在し、該指示されたデータアイテムに関連するデータアイテムを探索する探索工程と、

前記指示されたデータアイテムと前記探索工程で探索されたデータアイテムに対して前記特定プロセスによる排他的制御権を設定する設定工程とを備えることを特徴とする情報処理方法。

【請求項 2】 前記データは、仮想空間のコンピュータグラフィックスを生成する際に参照されるシーングラフデータベースであることを特徴とする請求項 1 記載の情報処理方法。

【請求項 3】 前記設定工程は、前記指示されたデータアイテムと前記探索工程で探索された全てのデータアイテムに関して、他のプロセスによって排他的制御権が設定されていない場合に、前記特定プロセスによる排他的制御権を設定することを特徴とする請求項 1 に記載の情報処理方法。

【請求項 4】 前記設定工程は、前記指定されたデータアイテムと前記探索工程で探索されたデータアイテムのうち、他のプロセスによって排他的制御権が設定されていないデータアイテムに対して、前記特定プロセスによる排他的制御権を設定することを特徴とする請求項 1 に記載の情報処理方法。

【請求項 5】 排他的制御権を解放するべく所望のデータアイテムを指示する第 2 指示工程と、

前記第 2 指示工程で指示されたデータアイテムに関して前記探索工程によって

関連するデータアイテムを探索し、指示されたデータアイテムと探索されたデータアイテムに対して前記特定プロセスによる排他的制御権を解放する第 1 解放工程とを更に備えることを特徴とする請求項 1 乃至 4 のいずれかに記載の情報処理方法。

【請求項 6】 前記第 2 指示工程で指示されたデータアイテムに関して前記探索工程によって関連するすべてのデータアイテムを探索し、該すべてのデータアイテムと前記指示されたデータアイテムが他のプロセスによって排他的制御権が設定されていない場合に、前記特定プロセスによる排他的制御権を解放する第 2 解放工程とを更に備えることを特徴とする請求項 1 乃至 4 のいずれかに記載の情報処理方法。

【請求項 7】 前記設定工程は、前記指示されたデータアイテムと前記探索工程で探索された全てのデータアイテムに関して他のプロセスによって排他的制御権が設定されていない場合に前記特定プロセスによる排他的制御権を設定する第 1 の設定モードと、前記指定されたデータアイテムと前記探索工程で探索されたデータアイテムのうち、他のプロセスによって排他的制御権が設定されていないデータアイテムに対して、前記特定プロセスによる排他的制御権を設定する第 2 の設定モードを有し、

前記第 1 及び第 2 の設定モードのうちの指定されたモードで排他的制御権の設定が実行されることを特徴とする請求項 1 に記載の情報処理方法。

【請求項 8】 前記第 1 及び第 2 の設定モードのうちの所望のモードをユーザによって指定可能であることを特徴とする請求項 7 に記載の情報処理方法。

【請求項 9】 前記第 1 及び第 2 の設定モードのうちのいずれのモードを適用するかが前記複数のデータアイテムの各々に付与されていることを特徴とする請求項 7 に記載の情報処理方法。

【請求項 1 0】 前記複数のデータアイテムの各々に対して排他的制御権の設定可能数が設定されており、

前記設定工程は、この設定可能数を限度に排他的制御権を設定することを特徴とする請求項 1 に記載の情報処理方法。

【請求項 1 1】 情報伝達媒体を介して通信可能な複数のプロセスによって

、複数のデータアイテムで構成されたデータを共有するシステムにおいて、特定プロセスによるデータアイテムの排他的制御権を設定するための情報処理装置であって、

前記複数のデータアイテムの階層構造情報を保持する保持手段と、

排他的制御権を設定するべく所望のデータアイテムを指示する第 1 指示手段と

、  
前記保持手段に保持された階層構造情報に基づいて、前記第 1 指示手段で指示されたデータアイテムよりも下位の階層に存在し、該指示されたデータアイテムに関連するデータアイテムを探索する探索手段と、

前記指示されたデータアイテムと前記探索手段で探索されたデータアイテムに対して前記特定プロセスによる排他的制御権を設定する設定手段とを備えることを特徴とする情報処理装置。

【請求項 1 2】 請求項 1 乃至 1 0 のいずれかに記載の情報処理方法をコンピュータに実行させるための制御プログラム。

【請求項 1 3】 請求項 1 乃至 1 0 のいずれかに記載の情報処理方法をコンピュータに実行させるための制御プログラムを格納した記憶媒体。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、複数のプロセス間でデータを共有し操作する技術に関する。特に 3 次元仮想空間中の物体を記述するデータを保持するデータベースをネットワークを介して複数のプロセス間での排他制御に好適な技術に関する。

【0 0 0 2】

【従来の技術】

異なるコンピュータ端末間で 3 次元仮想空間を共有する技術は、遠隔会議システム、ネットワーク型対戦ゲーム、協調型デザインシステムなどを実現するために必須である。

【0 0 0 3】

このような 3 次元仮想空間を共有するシステムでは、各端末上で仮想空間の動



画像をコンピュータグラフィクスにより描画するために仮想空間データが用いられる。このような仮想空間データを管理するための方式として以下のようなものがある。一つは、サーバ上に仮想空間データを持ち、各端末がネットワークを介してサーバ上の仮想空間データを参照し、CG画像を生成する方式である。また、別の方式として、各々の端末が仮想空間データのコピーを持ち、それを参照することによってCG画像を生成する方式がある。各々の端末が仮想空間データのコピーを持つ方式の場合、あるコンピュータ端末上で仮想空間に対して何らかの操作、例えば仮想物体の移動や回転が行われた場合には、その操作に関する情報が他の端末にネットワーク経由で伝送され、各端末の持つデータベースに反映される。こうして、各コンピュータ端末におけるデータベースの同一性が保たれる。

#### 【0004】

しかしながら、いずれの方式においても、同一の仮想空間データに対して、各端末より同時または時間的に極めて近いタイミングで操作が行われた場合には、意図しない結果を引き起こしたり、仮想空間の整合が取れなくなる可能性がある。

#### 【0005】

この問題に対処するための方法として、排他的制御権という概念が知られている。これは、ある操作対象を操作しようとする端末は、あらかじめ、その操作対象を独占的に操作するための権利（排他的制御権）を取得してから操作を行うものである。排他的制御権が設定された操作対象については、排他的制御権を取得した端末以外からは操作されることがない。すなわち、同時に当該操作対象を操作することができる端末は唯一であることが保証されるため、上記の問題点については解消される。

#### 【0006】

##### 【発明が解決しようとする課題】

一般に3次元空間中に配置される物体は階層的な相互関係を持っている。ある物体に対して移動させるなどの操作を行うと、その操作はその物体に属するすべての物体に対して波及する。例えば、物体の上に何か別の物体が載っているよう

な場合、下の物体を移動させると、通常は上に載っている物体もそれに合わせて、位置を変化させる。この物体間の階層的な相互関係を適切に表現するため、仮想空間データはN分木などの木構造、あるいは非再帰有向グラフなどのデータ構造によって記述されるのが一般的である。

#### 【0007】

しかしながら、従来は、仮想空間の階層関係とは無関係に、個々の物体に対して排他的制御権が設定されていた。そのため、物体の操作を行うためには、あらかじめ物体が属する仮想空間の階層関係を知った上で、当該物体に関連する他の物体についても個別に排他的制御権を取得する必要がある、大変煩わしいものであったため、改善が求められていた。

#### 【0008】

本発明は、上記の課題に鑑みてなされたものであり、仮想空間の階層構造に基づいた排他的制御権の取得方法を提供することにより、操作する側に仮想空間の構造を特別に意識させることなく、必要な排他的制御権を適切に取得できるようにすることを目的とする。

#### 【0009】

##### 【課題を解決するための手段】

上記の目的を達成するための本発明による情報処理方法は、

情報伝達媒体を介して通信可能な複数のプロセスによって、複数のデータアイテムで構成されたデータを共有するシステムにおいて、特定プロセスによるデータアイテムの排他的制御権を設定するための情報処理方法であって、

排他的制御権を設定するべく所望のデータアイテムを指示する第1指示工程と

前記複数のデータアイテムの階層構造情報に基づいて、前記第1指示工程で指示されたデータアイテムよりも下位の階層に存在し、該指示されたデータアイテムに関連するデータアイテムを探索する探索工程と、

前記指示されたデータアイテムと前記探索工程で探索されたデータアイテムに対して前記特定プロセスによる排他的制御権を設定する設定工程とを備える。

#### 【0010】

また、上記の目的を達成するための本発明による情報処理装置は以下の構成を備える。すなわち、

情報伝達媒体を介して通信可能な複数のプロセスによって、複数のデータアイテムで構成されたデータを共有するシステムにおいて、特定プロセスによるデータアイテムの排他的制御権を設定するための情報処理装置であって、

前記複数のデータアイテムの階層構造情報を保持する保持手段と、

排他的制御権を設定するべく所望のデータアイテムを指示する第1指示手段と

、  
前記保持手段に保持された階層構造情報に基づいて、前記第1指示手段で指示されたデータアイテムよりも下位の階層に存在し、該指示されたデータアイテムに関連するデータアイテムを探索する探索手段と、

前記指示されたデータアイテムと前記探索手段で探索されたデータアイテムに対して前記特定プロセスによる排他的制御権を設定する設定手段とを備える。

#### 【0011】

##### 【発明の実施の形態】

以下、添付の図面を参照して本発明の好適な実施形態を説明する。

#### 【0012】

##### 〔第1実施形態〕

第1実施形態では、本願発明による情報処理方法を、仮想空間の構造と属性を記述するシーンデータベースを複数の端末で共有する仮想空間共有システムに適用した実現形態について説明する。

#### 【0013】

本実施形態の仮想空間共有システムの全体的な構成を図1に示す。システムには1つのサーバプロセス101（以下、「サーバ」と略す）と、複数のクライアントプロセス103（以下、「クライアント」と略す）があり、情報伝達媒体としてのネットワーク102を介してデータを交換する。それぞれのクライアントは共通の仮想空間の構造と属性を記述するシーンデータベースを保持する。なお、本実施形態では、ネットワーク102はイーサネット（登録商標）上に構築したLANである。また、以下の説明では、サーバとクライアントとを総称して「

ホスト」と呼ぶことがある。なお、情報伝達媒体は有線でも無線でもかまわない。

#### 【0 0 1 4】

ここで、以下の説明で用いる「データベース操作」という用語について説明しておく。データベースの「操作」とは、共有データベースを持つ任意のプロセスにおけるデータベースの内容を書き換える処理のことを差す。このデータベース操作は「操作指令」と「操作実行」の2段階に分かれる。操作指令はデータベースの更新要求であり、実際の書き換えは行われない。データベースに対する実際の書き換えは、操作実行の段階で行われる。以下の説明では特に断らない限り、単に「操作」と記述すれば「データベース操作」のことを指すものとする。操作対象がデータベース以外の場合は、「ユーザによる対話デバイスの操作」などのように操作対象を明示するものとする。

#### 【0 0 1 5】

なお、操作指令はユーザが発生させることもあれば、プロセスが動作する過程で発生することもある。前者の例としては、ユーザがマウスなどの対話デバイスを操作して、仮想オブジェクトを移動させた際に操作指令が発生されることが挙げられる。一方、後者の例としては、シューティングゲーム・システムにおいて、ゲームプログラムが敵キャラクタをアルゴリズム的に移動・回転させた際に操作指令が発生されること等が挙げられる。

#### 【0 0 1 6】

さて、図2はデータベースを更新するための基本的な情報の流れを示すものである。図2で、A, B, C, D (2 0 1 ~ 2 0 4) はクライアント、X (2 0 5) はサーバである。いま、クライアントA (2 0 1) がシーンデータベースの操作指令を行ったとする。操作指令の内容はネットワーク経由でサーバに送信される(図2で(1)の矢印の過程)。サーバは、この操作指令を受けてクライアントA (2 0 1) から送信されたデータを、接続されているすべてのクライアントに配信する(図2で(2)の矢印の過程)。このような過程を経ることによって、非操作者のクライアントB (2 0 2)、C (2 0 3)、D (2 0 4) も操作指令の内容を知ることができる。

**【0017】**

次に、クライアントまたはサーバプロセスが動作する端末の構成を図3に示す。図3において、301はCPUであり、端末全体の動作を制御する。302はメモリであり、CPU301の動作に用いるプログラムやデータを格納する。303はバスであり、各構成モジュール間のデータ転送を司る。304はバス303と各種装置とのインタフェースである。305はネットワークと接続するための通信部、306はCPU301に読み込むプログラムやデータを格納する外部記憶装置、307および308はプログラムを起動したり、プログラムの動作を指定するための入力装置を構成するキーボードおよびマウス、309はプロセスの動作結果を表示するための表示部である。

**【0018】**

本実施形態のシステムでは、ある操作指令に対する操作実行をサーバからのイベントの配信を待って行うことで、各クライアントのデータベースの更新タイミングを同期させる。図4は、時間の進行に沿った処理手順を模式的に示したものである。図4で、「操作者」は操作指令を行ったクライアント、「非操作者」は操作者以外のクライアントを指す。処理の進行は以下の通りである。

**【0019】**

まず、手順T401で操作者クライアントがデータベースの操作指令を発する。この時点では、操作者クライアントのデータベースは更新されない。次に手順T402にて、T401の操作内容を示すイベントが生成され、T403でこのイベントがサーバに送信される。サーバは手順T404でイベントを受信すると、T405で当該サーバとの接続が確立しているすべてのクライアントに向けてイベントを送信する（接続の確立について図7のステップS702で後述する）。サーバより発行されたイベントを操作者クライアントが手順T406で受信し、その内容を分析して（T407）、データベースを書き換える（操作実行、T408）。イベントの受信から操作実行までの手順は、非操作者クライアントでも同様である（手順T409～T411）。図4に示したように、本実施形態のデータベース操作では、操作指令（T401）から操作実行（T408）までタイムラグがある。しかし、ネットワーク上の情報転送の条件が同様であれば、操

作者・非操作者クライアントともほぼ同時刻にデータベースが更新され、仮想空間の画像がクライアント間で同期することが期待できる。

#### 【0 0 2 0】

図 5 は、クライアントからサーバに対して、或いはサーバからクライアントに対して発行されるイベントの構成を示したものである。イベントのデータは複数のフィールドから構成される。クライアント ID 5 0 1 は、各クライアントを一意に識別するための番号で、クライアントがサーバとネットワーク接続を確立する際にサーバより割り当てられたものである（図 7 のステップ S 7 0 2 により後述する）。操作 ID 5 0 2 は操作を一意に識別するための番号で、各クライアントごとに操作指令を行った時点で割り当てられる。このクライアント ID 5 0 1 および操作 ID 5 0 2 の組を用いることにより、当該システム内部で発生したすべての操作指令を一意に特定することが可能である。

#### 【0 0 2 1】

エントリ ID 5 0 3 は、データベースの各エントリを一意に識別するための番号で、データをファイルからロードした際に割り当てられる。操作内容 5 0 4 は、操作指令の具体的内容であり、例えば CG オブジェクトの X 座標をある値に設定する操作であれば、X 座標属性を識別する番号と X 座標の設定値の組で表される。

#### 【0 0 2 2】

なお、操作イベントを受信した時点で先に発行された別の操作イベントに対応した操作を実行中の場合には、後に受信した操作イベントは操作キューに保持され、先の操作実行が終了した後に順次実行される。図 6 は操作キューを示す図である。操作キューは、イベントの受信または実行処理を待機している操作指令のリストである。ある時点で待機状態にある操作指令が m 個あるとすると、操作キューは図 6 のように操作指令に関する m 個の情報（キューアイテム）を指令の発生順に並べたものとなる。

#### 【0 0 2 3】

次に、クライアント、サーバのそれぞれについて、データベース操作の処理の流れを詳しく説明する。

**【 0 0 2 4 】**

図 7 は、クライアントの処理におけるデータベース操作の全体的な流れを示すフローチャートである。クライアントが起動されると、ステップ S 7 0 1 にて仮想空間を記述するファイルを外部記憶装置 3 0 6（図 3）より読み込み、メモリ 3 0 2（図 3）上にシーングラフデータベースを構築する。クライアント間でのデータの同一性を維持するため、ここでロードするファイルはクライアント間で同じ内容を持つとする。次に、サーバとのネットワーク接続を確立する（ステップ S 7 0 2）。このとき、サーバから各クライアントを識別するためのクライアント ID 5 0 1（図 5）が付与される。なお、クライアントとサーバ間のデータ交換は、TCP/IP プロトコルを用いた 1 対 1 のソケット通信にて行う。したがって、サーバは少なくともクライアントの数だけのソケット通信路を持つことになる。

**【 0 0 2 5 】**

続いて、ステップ S 7 0 3 でプロセスを分岐し、操作指令を処理するための操作処理（ステップ S 7 0 4）およびサーバから受信するイベントを処理するための受信イベント処理（ステップ S 7 0 6）を起動する。なお、図 7 には図示していないが、シーングラフデータベースを参照して、仮想空間の CG 画像を生成するプロセスも起動する。この処理は、公知の CG 画像生成方法と同様であるのでその詳細な説明は省略する。また、操作処理および受信イベント処理については、後に詳しく説明する。

**【 0 0 2 6 】**

操作処理および受信イベント処理について、それぞれステップ S 7 0 5 およびステップ S 7 0 7 でプロセスの終了する指示があったかどうか判定され、終了指示がなければステップ S 7 0 4 およびステップ S 7 0 6 に戻って操作指令および受信イベントを継続して処理する。一方、終了指示があればステップ S 7 0 8 でプロセスを統合し、サーバとのネットワーク接続を切断（ステップ S 7 0 9）した上で、データベースの内容を外部記憶装置 3 0 6（図 3）にファイルとしてセーブ（ステップ S 7 1 0）して、すべての処理を終了する。

**【 0 0 2 7 】**

ここで、ステップ S 7 0 4 の操作処理について図 8 を参照して詳しく説明する。初めに、ステップ S 8 0 1 にて操作指令の内容を入力する (T 4 0 1)。このとき、操作指令を一意に特定する操作 I D 5 0 2 (図 5) を決定する。次に、ステップ S 8 0 2 にて、ステップ S 8 0 1 で入力した操作の内容に基づいて、イベントデータ (図 5 参照) を生成する (T 4 0 2)。そして、ステップ S 8 0 3 に進み、生成したイベントデータをサーバに向けて発信する (T 4 0 3)。

#### 【 0 0 2 8 】

続いて、ステップ S 7 0 6 (図 7) の受信イベント処理について詳しく説明する。図 9 は、受信イベント処理の流れを示した図である。受信イベントは、通信部 3 0 5 を介してサーバより受信され、不図示の受信イベントバッファに入力される。

#### 【 0 0 2 9 】

ステップ S 9 0 1 では、受信イベントバッファを検索する。続くステップ S 9 0 2 で、受信イベントバッファにイベントが入力されているかどうか判定し、受信イベントがあればステップ S 9 0 3 に進む。一方、受信イベントが存在しないと判定されれば、処理を終了する。受信イベントがある場合、そのイベントデータを解析し、操作指令の内容を抽出する (ステップ S 9 0 3、T 4 0 6、T 4 0 7、T 4 0 9、T 4 1 0)。次に、ステップ S 9 0 4 でイベントに記述された操作を実行する (T 4 0 8、T 4 1 1)。具体的には、エントリ I D 5 0 3 によって指定されたエントリに関して、操作内容 5 0 4 の内容に従って設定値を変更する等の操作を実行する。

#### 【 0 0 3 0 】

以上、クライアント側プロセスの流れを説明した。続いて、サーバ側プロセスの流れを図 1 0 を用いて詳しく説明する。

#### 【 0 0 3 1 】

最初にステップ S 1 0 0 1 で、クライアントからの接続要求を受け付け、通信を確立する。このとき、通信を確立した各クライアントにクライアント I D 5 0 1 を通知する。次に、受信イベントバッファを検索し (ステップ S 1 0 0 2)、受信イベントが存在するかどうか判定する (ステップ S 1 0 0 3)。受信イベン



トがあれば、ステップ S 1 0 0 4 に進み、なければステップ S 1 0 0 5 に進む。ステップ S 1 0 0 4 では、接続している各クライアントに当該イベントを送信する (T 4 0 4、T 4 0 5)。ステップ S 1 0 0 5 では、ユーザからの指令などによりサーバ処理を終了するかどうかを判定し、終了する場合はステップ S 1 0 0 6 に進み、終了しない場合はステップ S 1 0 0 2 に戻る。ステップ S 1 0 0 6 では、接続している各クライアントに処理を終了することを通知し、次にクライアントとの接続を終了して (ステップ S 1 0 0 7) から処理を終了する。

### 【 0 0 3 2 】

次に、本実施形態でクライアントが読み込むデータベース記述ファイルの構成について説明する。

### 【 0 0 3 3 】

図 1 1 はエントリ数が N のデータのファイルフォーマットを模式的に示したものである。各エントリの属性は、スタートデリミタおよびエンドデリミタで囲まれたフィールドに記述される。図 1 1 で、1 番目、2 番目、…、N 番目のエントリのスタートデリミタはそれぞれ 1 1 0 1、1 1 0 6、1 1 0 9 で示されている。一方、それぞれのエンドデリミタは 1 1 0 5、1 1 0 8、1 1 1 1 である。エントリの属性は、1 1 0 2 ~ 1 1 0 4 のように、各属性を識別するための識別子および属性値の組で記述される。

### 【 0 0 3 4 】

また、上記の説明では、サーバ・クライアント間のデータ通信を 1 対 1 の T C P / I P ソケットにて行うとしているが、ブロードキャストあるいはマルチキャストによりイベントを交換してもよい。また、通信プロトコルは T C P / I P に限らない。また、通信媒体としてはイーサネット (登録商標) を利用しているが、U S B や Firewire など他の情報伝達媒体でも構わない。更に、ネットワークの形態も L A N だけでなく、W A N 経由の接続でもよいし、L A N と W A N を併用しても構わない。

### 【 0 0 3 5 】

また、サーバの数は、1 つの共有データベースシステムにつき 1 つに限定する必要はなく、複数のサーバが存在してもよい。この場合、各サーバに処理を割り

当てる形態は任意である。例えば、クライアントごとに異なるサーバを用いてもよいし、あるいはデータベースのエントリごとにサーバを設けてもよい。

#### 【0 0 3 6】

更に、端末に対するプロセスの割当は1端末1プロセスに限定されるものではなく、1つの端末上で複数のプロセスを動作させることができる。その場合、プロセスの種別の組み合わせは任意であり、クライアントのみ、サーバのみ、クライアントおよびサーバの、いずれの組み合わせでもシステムは動作する。なお、同じ端末上のプロセスは、ネットワークに替えて、共有メモリを経由してデータを交換してもよい。

#### 【0 0 3 7】

また、イベントはサーバが仲介して各端末に配信されるが、データ操作を行った端末が直接に他の端末にイベントを送信してもよい。

#### 【0 0 3 8】

また、サーバプロセスにおいては、情報伝達媒体を介して通信可能に接続された複数のプロセスによって共有される共有データを該複数のプロセスの各々が保持して利用するシステムにおいて、各プロセスが保持する共有データの同一性を維持するために、複数のクライアントプロセスとの接続を確立し、これら複数のクライアントプロセスより共有データの変更に関するイベントを受信する。そして、受信したイベントを、上記複数のクライアントプロセスに対して発行する。

#### 【0 0 3 9】

次に、本実施形態のシステムにおいて仮想空間を表現するシーンデータベースの構造について説明する。

#### 【0 0 4 0】

本実施形態におけるシステムでのシーンデータベースは「シーングラフ」と呼ばれる木構造、あるいは非再帰有向グラフ構造をとる。シーングラフは、仮想空間の構造的な依存関係を記述するためのデータ構造である。図12では床1201の上にテーブル1202があり、さらにテーブル1202の上に箱1203が載っている仮想空間の例を示している。また、図13では、図12の仮想空間を表現するシーングラフの例を示している。図13中の丸印は木構造の節（ノード

)を、直線は枝を意味する。以後、添付の図面において、特に断らない限りは、シーングラフのノードは丸印で、枝は直線で表現されるものとする。なお、各ノードは操作対象のデータアイテムを表す。

#### 【0041】

図13のシーングラフでは、木構造の根に床1201が配置され、その子としてテーブル1202が、さらにテーブル1202の子として箱1203が配置されている。このようなデータ構造を用いることにより、仮想空間において床1201を移動させると、あたかもテーブル1202が床1201に、箱1203がテーブル1202の上に実際に載っているかのように、テーブル1202、箱1203も床1201に追従して移動させることができる。

#### 【0042】

ところで、本実施形態のシステムにおいて、データベース中のある操作対象に対して、複数のクライアントが同時に操作指令を発行した場合には、操作者が意図しない結果を引き起こす可能性が生じる。例えば、不図示のクライアント1、クライアント2およびサーバがデータベースを共有しており、クライアント1が不図示の操作対象TをX軸の正方向に1だけ平行移動させる操作指令Aを、クライアント2が操作対象TをX軸の負方向に1だけ平行移動させる操作指令Bをほぼ同時に発行する場合を考える。サーバではクライアント1からの操作指令A、クライアント2からの操作指令Bを受け取り、各操作指令がサーバに到着した順に操作対象Tに対して操作を実行する。操作指令A、操作指令Bの順に実行した場合には、操作対象Tは、操作指令AによってX軸の正方向に1だけ平行移動し、操作指令BによってX軸の負方向に1だけ平行移動する。この結果、操作指令Bが終了した時点で操作対象Tは、操作対象Aが実行される直前の状態と等しくなる。操作指令B、操作指令Aの順に実行した場合も、操作指令Aが終了した時点で操作対象Tは、操作対象Bが実行される直前の状態と等しくなる。操作指令A、操作指令Bは微小時間内に実行されるため、各クライアントまたはサーバでは、操作対象Tはまったく移動しないか、または、振動しているように観察される。すなわち、この場合、クライアント1は操作対象Tが操作指令AによってX軸の正方向に1だけ平行移動することを意図しており、また、クライアント2は

操作指令 B によって X 軸の負方向に 1 だけ平行移動することを意図しているため、実際の操作結果はクライアント 1, 2 のどちらにとっても意図しないものとなっている。

#### 【 0 0 4 3 】

上記のような問題を回避するため、ある操作対象を操作しようとするホストは、事前に当該操作対象を独占的に操作する権利を取得し、その後に当該操作対象の操作指令を発行する。この権利を「排他的制御権」と呼ぶ。排他的制御権が設定されている操作対象については、排他的制御権を取得しているホスト以外からの操作指令は実行されず、排他的制御権を取得しているホストのみが当該操作対象の操作を行うことが可能となる。また、あるホストが排他的制御権を有している操作対象に関しては、当該操作対象の排他的制御権を取得したホストが排他的制御権を解放するまでの間、他のホストが排他的制御権を取得することはできない。排他的制御権を取得したホストが排他的制御権を解放した場合、当該操作対象は排他的制御権が設定されていない状態となり、すべてのホストの操作が許可されることになる。

#### 【 0 0 4 4 】

ホストが排他的制御権の取得を行うときには、「排他的制御権取得」の操作要求を発行する。具体的には、クライアントが 3 次元空間の表示画面上より所望の操作対象を選択して排他的制御権の取得要求を操作イベントとして発行する。この取得要求はサーバによって受信され各クライアントへ通知される。当該取得要求の発行元のクライアントでは以下に説明するように選択された操作対象とこれに関連する対象（データアイテム）について排他的制御権の取得が行われる。排他的制御権の取得結果サーバを介して全クライアントに通知される。

#### 【 0 0 4 5 】

通常、排他的制御権は操作対象ごとに独立に設定される。そのため、操作対象ごとにそれぞれ異なるホストが排他的制御権を取得することが可能である。例えば図 1 4 のシーングラフにおいて、クライアント 1 が操作対象 1 4 0 1 の排他的制御権を、クライアント 2 が操作対象 1 4 0 2 の排他的制御権を同時に取得することが可能である。

**【0046】**

上記の方法では、操作対象ごとに完全に独立して排他的制御権を取得することができるため、階層構造を持つシーングラフに適用した場合に問題が生じることがある。例えば図13のシーングラフにおいて、図示しないホストAが床1201の排他的制御権を取得した場合、図示しない別のホストBがテーブル1202の排他的制御権を、さらに図示しない別のホストCが箱1203の排他的制御権を取得することができる。図13のシーングラフでは、テーブル1202は床1201に、箱1203はテーブル1202と床1201に属している。そのため、床1201の排他的制御権を取得しているホストAが床1201を操作すると、テーブル1202、箱1203にまでその操作の影響が波及する場合がある。しかし、ホストAは床1201のみの排他的制御権しか取得していないため、テーブル1202、箱1203に関しては、他のホストによって操作されて意図しない操作結果となる可能性がある。

**【0047】**

これを回避するためには、ホストAはテーブル1202、箱1203の排他的制御権についても床1201の排他的制御権と同時に取得しておかなければならない。上記の方法では、ホストAはシーングラフの構造に基づいて、操作するノードに階層的に関連したノードの排他的制御権を個別に取得する必要がある。すなわち、各ホストは操作するシーングラフの階層構造を正確に把握した上で、操作対象となる全ノードについて排他的制御権を取得しなければならない。この作業は、シーングラフの規模に比例して煩雑なものとなる。

**【0048】**

本実施形態では、シーングラフ中で上位のノードに該当する操作対象の排他的制御権を取得した場合には、その下位の全ノードに該当する操作対象について排他的制御権を取得する。すなわち、ある操作対象の排他的制御権を取得すると、当該操作対象を根とする部分木（サブシーングラフ）に含まれるすべての操作対象の排他的制御権をも取得する。以後、本明細書では、シーングラフ中のノードに該当する操作対象について排他的制御権を取得することを、単に「ノードの排他的制御権を取得する」と表現する。また、サブシーングラフに含まれるすべて

の操作対象の排他的制御権を取得することを、単に「サブシーングラフの排他的制御権を取得する」と表現する。本実施形態によれば、操作対象以下のサブシーングラフのすべてのノードの排他的制御権が自動的に取得される。このため、操作を行うホストは操作対象の情報のみを所持していればよく、シーングラフの構造を正確に把握している必要はない。

#### 【0 0 4 9】

図 1 5 は、上記に述べた排他的制御権を取得する処理の流れを示している。ステップ S 1 5 0 0 では、指定の操作対象について、排他的制御権を取得する処理を開始する。ステップ S 1 5 0 5 では、指定のホスト（排他的制御権の取得要求元のクライアント）以外のホストが自ノード（その階層における排他的制御権の設定対象ノード）に対して排他的制御権を取得しているかを確認する。排他的制御権が取得されていた場合には、ステップ S 1 5 0 8 に進み、処理の呼び出し元（再帰処理を行なう上位の階層のノードに対する処理）に「失敗」を示す情報を通知して本処理を終了する。当該自ノードがホストより排他的制御権の取得を要求されたノードの場合は、当該ホストに対して取得の失敗を返すことになる。

#### 【0 0 5 0】

取得されていなかった場合にはステップ S 1 5 0 1 に進む。ステップ S 1 5 0 1 では、当該自ノードのすべての子ノードについて、排他的制御権を取得する処理が完了したかを判定する。すべての子ノードについて排他的制御権を取得する処理が完了したか、または、子ノードが存在しない場合にはステップ S 1 5 0 6 に進む。そうでない場合には、ステップ S 1 5 0 2 に進む。ステップ S 1 5 0 2 では、各子ノードについて、排他的制御権を取得する処理を行う。この処理は再帰的に行われる。すなわち、処理対象を子ノードとして、新たに排他的制御権を取得するための処理を呼び出す。より具体的には、現時点までの処理結果を保存してから処理を中断し、処理対象とした子ノードに対する処理（当該子ノードを自ノードとした処理）を新たにステップ S 1 5 0 0 から開始する。

#### 【0 0 5 1】

ステップ S 1 5 0 3 では、ステップ S 1 5 0 2 で呼び出した新たな排他的制御権を取得するための処理が成功したか失敗したかを判定する。成功した場合には

、次の子ノードを処理するべくステップS1501に戻る。失敗した場合には、ステップS1504に進む。ステップS1504では、処理の呼び出し元に「失敗」を示す情報を通知し、ステップS1509に進む。

#### 【0052】

ステップS1501において未処理の子ノードが存在しないと判定されたならば、ステップS1506で自ノードに排他的制御権を設定し、データベースに排他的制御権を取得したホストのホストIDを書き込む。そして、ステップS1507において、処理の呼び出し元に「成功」を示す情報を通知し、ステップS1509に進む。一方、ステップS1508では、処理の呼び出し元に「失敗」を示す情報を通知し、ステップS1509に進む。ステップS1509では、排他的制御権の取得処理を終了する。再帰呼び出しの過程で中断している処理がある場合には、中断している処理を再開する。

#### 【0053】

例えば、図23に示すようなシーングラフがあった場合、処理は、O→A→C→D→B→Eの順に行われる。Oに対する処理が開始されると、未処理の子ノードが存在するので処理はステップS1502に進み、Oに対する処理を中断してAに対する処理がステップS1500より開始される。同様に、Aに対する処理を行うと、ステップS1502に進み、Aに対する処理が中断されて、Cの処理がステップS1500から開始される。Cには子ノードがないので、Cに対して排他的制御権が設定されていなければステップS1501→S1506→S1507→S1509と処理が進み、Cの処理に成功して終了することになる。この結果、呼び出し元の処理（Cの上の階層のAに対する処理）が再開され、ステップS1503からステップS1501へ戻ることになる。

#### 【0054】

Aには、未だ未処理の子ノードDが存在するので、ステップS1502へ進み、Dの処理がステップS1500から開始される。Cと同様にDの処理に成功すると、再びAの処理が再開される。今度は全ての子ノードについて処理が終了しているので、ステップS1501からステップS1506へ進み、Aに対して排他的制御権が設定され、処理が終了する。Aに対する処理が終了すると、中断し

ていた O に対する処理が再開されるが、O には未処理の子ノード B が存在するので、再び処理が中断され、B に対する処理が開始されることになる。以上のようにして、O、A、C、D、B、E の全てのノードについて排他的制御権の設定に成功した場合に、O に対する排他的制御権が設定されることになる。

#### 【0 0 5 5】

一方、排他的制御権を解放する場合については、あるサブシーングラフの排他的制御権を取得しているとき、解放すべきサブシーングラフの根となるノードの排他的制御権の解放を指示することにより、その下位の全ノードの排他的制御権が解放される。

#### 【0 0 5 6】

図 1 6 は、排他的制御権を解放する処理の流れを示している。ステップ S 1 6 0 0 では、指定の操作対象について、排他的制御権を解放する処理を開始する。ステップ S 1 6 0 1 では、当該ノードのすべての子ノードについて、排他的制御権を解放する処理が完了したかを判定する。すべての子ノードについて排他的制御権を解放する処理が完了したか、または、子ノードが存在しない場合にはステップ S 1 6 0 3 に進む。そうでない場合には、ステップ S 1 6 0 2 に進む。

#### 【0 0 5 7】

ステップ S 1 6 0 2 では、各子ノードについて、排他的制御権を解放する処理を行う。この処理は再帰的に行われる。すなわち、処理対象を子ノードとして、新たに排他的制御権を解放するための処理を呼び出す。現時点までの処理結果を保存してから処理を中断し、子ノードに対する処理を新たにステップ S 1 6 0 0 から開始する。

#### 【0 0 5 8】

ステップ S 1 6 0 3 では、自ノードに対して排他的制御権を設定したホストが、指定のホストと一致するかを確認する。なお、本実施形態では排他的制御権を解放できるのは対象ノードに排他的制御権を設定したホストのみとしている。すなわち、排他的制御権を設定したときに、そのノード（のデータ領域）に排他的制御権を設定したホストの ID を登録しておき、解放時には保持されている ID と解放処理を呼び出しているホストの ID とを照合し、これが一致したときのみ



解放を行う。排他的制御権が設定されていない、あるいはホストが一致しない場合には、ステップ S 1 6 0 5 に進む。それ以外の場合にはステップ S 1 6 0 4 に進む。

#### 【0 0 5 9】

ステップ S 1 6 0 4 では、自ノードに対して設定されている排他的制御権を解放し、データベースに記述されている排他的制御権を取得したホストのホスト ID を削除する。ステップ S 1 6 0 5 では、排他的制御権の解放処理を終了する。再帰呼び出しの過程で中断している処理がある場合には、中断している処理を再開する。

#### 【0 0 6 0】

例えば、図 1 7 のシーングラフにおいて、黒で示されるノードの排他的制御権を取得しようとしている場合、黒に加えて、灰色で示される下位ノードの排他的制御権も同時に取得する。同様に、黒で示されるノード 1 7 0 1 の排他的制御権を解放すると、ノード 1 7 0 1 に加えて、灰色で示される下位ノード（1 7 0 2）の排他的制御権も同時に解放する。

#### 【0 0 6 1】

ここで、当該操作対象を根とするサブシーングラフのうち、一つでも排他的制御権を取得することができないノードが存在する場合には、そのサブシーングラフのすべてのノードの排他的制御権を取得できない。例えば、図 1 8 のシーングラフにおいて、図示しないホスト A が黒の丸で示されるノード 1 8 0 1 の排他的制御権を取得しようとしており、既に灰色の四角形で示されるノードの排他的制御権が図示しないホスト B によって取得されているような場合には、ホスト A は灰色の四角印の下位ノードの排他的制御権を取得することができない。よって、ホスト A は黒の丸印で示されるノード 1 8 0 1 の排他的制御権についても取得することができない。すなわち、あるノードの排他的制御権を取得する場合には、当該ノードを根とするサブシーングラフ中の全てのノードの排他的制御権が取得可能でなければならない。この条件が満たされない場合には、当該ノードの排他的制御権を取得することはできない。

#### 【0 0 6 2】

また、排他的制御権を取得したサブシーングラフについて、その一部を解放することも可能である。図 1 9 は、図示しないホスト A が黒で示されるノード以下のサブシーングラフ 1 9 0 1 の排他的制御権を取得している様子を示している。すなわち、ホスト A は黒および灰色で示されているノード群の排他的制御権を取得している。一方、図 2 0 は、図 1 9 の状態から、ホスト A が黒で示されるノード 2 0 0 1 以下のサブシーングラフの排他的制御権を解放した状態を示している。すなわち、図 1 9 の状態において、ホスト A がノード 2 0 0 1 を指定して排他的制御権の解放を指示すると、図 2 0 に示されるようにノード 2 0 0 1 以下のサブシーングラフの排他的制御権が解放される。

#### 【 0 0 6 3 】

また、排他的制御権を取得したサブシーングラフの一部を解放し、他のホストがその部分の排他的制御権を取得することも可能である。図 2 1 では、図 2 0 の状態から、上述したホスト B が黒の四角印で示されるノード 2 0 0 1 以下のサブシーングラフ 2 1 0 1 の排他的制御権を取得した状態を示している。図 2 1 では、灰色の丸印のノード群 2 1 0 2 に関してホスト A が、黒および灰色の四角印のノード群 2 1 0 1 に関してホスト B が排他的制御権を取得している状態を示している。

#### 【 0 0 6 4 】

例えば図 1 2 のシーンにおいて、不図示のホスト A が床 1 2 0 1 の排他的制御権を取得したとする。このとき、テーブル 1 2 0 2 および箱 1 2 0 3 は床 1 2 0 1 のサブシーングラフであるので（図 1 3）、テーブル 1 2 0 2 および箱 1 2 0 3 の排他的制御権もホスト A によって同時に取得される。次に、ホスト A はテーブル 1 2 0 2 の排他的制御権を解放し、不図示のホスト B がテーブル 1 2 0 2 の排他的制御権を取得する。このとき、ホスト A は同時に箱 1 2 0 3 の排他的制御権も解放することになるので、ホスト B は同時にテーブル 1 2 0 2 と箱 1 2 0 3 の排他的制御権を取得することになる。この結果、ホスト A は床 1 2 0 1 の排他的制御権を、ホスト B はテーブル 1 2 0 2 と箱 1 2 0 3 の排他的制御権を取得している状態となる。

#### 【 0 0 6 5 】

また、当該ノードを根とするサブシーングラフ中に、他のホストが排他的制御権を取得しているノードが存在した場合には、サブシーングラフ中の全てのノードの排他的制御権を解放できないようにすることも可能である。すなわち、サブシーングラフ中のいずれかのノードについて、他のホストが排他的制御権を取得している場合には、当該ホストが当該ノードの排他的制御権を解放しない限りは、その上位のノードあるいはサブシーングラフの排他的制御権を取得しているホストは、当該ノードあるいはサブシーングラフの排他的制御権を解放できないようにすることができる。

#### 【0066】

上述したように、従来例においては個別にノードの排他的制御権を設定する。したがって、従来例においても排他的制御権を取得したノードの子ノードまたはサブシーングラフを、他のホストに操作させるようにすることも可能ではある。しかしながら、その手順は本実施形態よりも煩雑となる。例えば上記の場合、従来例ではホストAはテーブル1202および箱1203の排他的制御権を解放し、ホストBはテーブル1202および箱1203の排他的制御権を取得する必要がある。このため、排他的制御権を解放・取得の操作を行う回数の合計は4回となる。一方、本実施形態の場合、ホストAはテーブル1202の排他的制御権を解放し、ホストBはテーブル1202の排他的制御権を取得すれば済むので、排他的制御権を解放・取得する回数の合計は2回となる。この差は、シーングラフの規模が大きくなり、シーングラフの構造が複雑になるほど顕著に現れる。

#### 【0067】

なお、上記の説明では、3次元仮想空間データを共有するとしているが、データベースの内容はこれに限らず任意であることはいうまでもない。また、データベース全体を共有する場合のみならず、その一部のみを共有する場合にも上記実施形態で説明した方法が適用可能であることは言うまでもない。これらの特徴は、以下の実施形態についても同様である。

#### 【0068】

以上のように、第1実施形態によれば、情報伝達媒体を介して接続された複数のプロセスによってデータを共有し、データは単数または複数の階層構造を持つ

部分データから構成され、部分データは単数または複数のデータアイテムから構成されるシステムにおいて、データアイテムについて特定のプロセスからの操作を許可し、当該プロセス以外のプロセスからの操作を拒否する排他的制御権を設定するための情報処理方法が開示され、排他的制御権を設定する対象ノードを指定すると、当該シーングラフの構造に基づき、対象ノード以下のサブシーングラフの排他的制御権をも同時に取得される。また、排他的制御権を取得した対象ノードを解放するときにも、シーングラフの構造に基づき、対象ノード以下のサブシーングラフの排他的制御権をも同時に解放される。このため、排他的制御権の取得、解放が容易になる。

#### 【0 0 6 9】

さらに、第 1 実施形態によれば、排他的制御権を取得したサブシーングラフの一部について排他的制御権を解放し、解放した排他的制御権を他のプロセスに取得させることが可能となる。柔軟な排他的制御権の設定を容易な操作で行なえる。

#### 【0 0 7 0】

##### 〔第 2 実施形態〕

第 1 実施形態では、操作対象を根とするサブシーングラフのうち、一つでも排他的制御権を取得することができないノードが存在する場合には、そのサブシーングラフのすべてのノードの排他的制御権を取得できない。第 2 実施形態では、そのような場合に、サブシーングラフ中で排他的制御権を取得可能ノードについて排他的制御権を取得するようにする。

#### 【0 0 7 1】

図 2 2 は、第 2 実施形態による排他的制御権を取得する処理の流れを示している。ステップ S 2 2 0 0 では、指定の操作対象について、排他的制御権を取得する処理を開始する。ステップ S 2 2 0 1 では、当該ノードのすべての子ノードについて、排他的制御権を取得する処理が完了したかを判定する。すべての子ノードについて排他的制御権を取得する処理が完了したか、または、子ノードが存在しない場合にはステップ S 2 2 0 3 に進む。そうでない場合には、ステップ S 2 2 0 2 に進む。

**【 0 0 7 2 】**

ステップ S 2 2 0 2 では、各子ノードについて、排他的制御権を取得する処理を行う。この処理は再帰的によって行われる。すなわち、処理対象を子ノードとして、新たに排他的制御権を取得するための処理を呼び出す。現時点までの処理結果を保存してから処理を中断し、子ノードに対する処理を新たにステップ S 2 2 0 0 から開始する。

**【 0 0 7 3 】**

ステップ S 2 2 0 3 では、指定のホスト以外のホストが自ノードに対して排他的制御権を取得しているかを確認する。排他的制御権が取得されていた場合には、ステップ S 2 2 0 5 に進み、取得されていなかった場合にはステップ S 2 2 0 4 に進む。ステップ S 2 2 0 4 では、自ノードに排他的制御権を設定し、データベースに排他的制御権を取得したホストのホスト I D を書き込む。

**【 0 0 7 4 】**

ステップ S 2 2 0 5 では、排他的制御権の取得処理を終了する。再帰呼び出しの過程で中断している処理がある場合には、中断している処理を再開する。

**【 0 0 7 5 】**

第 2 実施形態では、子ノードが排他的制御権を取得する処理（ステップ S 2 2 0 2）の後に、その処理が成功したか否かの判定を行わない。すなわち、子ノードが排他的制御権を取得できなかった場合においても、そこで処理を停止することなく、サブシーングラフの全ノードに対して排他的制御権の取得を試みる。なお、第 2 実施形態における、排他的制御権の解放の手順については、第 1 実施形態と同じである。

**【 0 0 7 6 】**

以上のように、第 2 実施形態によれば、操作対象を根とするサブシーングラフのうち、排他的制御権を取得することができないノードが存在する場合においても、サブシーングラフ中で排他的制御権を取得できるノードについてのみ、排他的制御権を取得することが可能となる。

**【 0 0 7 7 】**

〔他の実施形態〕

なお、第1実施形態と第2実施形態における排他的制御権の取得方法のうちいずれかの取得方法を選択して、排他的制御権の取得処理を起動できるようにしてもよい。

#### 【0078】

また、第1実施形態、第2実施形態における排他的制御権の取得方法のうちのいずれの取得方法を、適用するかをあらかじめノードごとに設定しておき、排他的制御権の取得処理が起動されると、当該ノードに設定された取得方法を用いて処理を実行するようにしてもよい。

#### 【0079】

また、上記各実施形態においては、あるノードの排他的制御権を同時に取得できるホストはただ一つとしているが、各ノードについて、あらかじめ定められた数だけの複数のホストが排他的制御権を取得することを可能にしてもよい。この場合において、排他的制御権を取得した複数のホストが同時に操作を行ったときには、排他的制御は行われなため、先述したように、意図しない操作結果を引き起こす可能性がある。しかし、同一ノードに同時に操作を行わないことが保証されるホストのみが排他的制御権を取得するようにすることで、そのような問題を回避することが可能である。

#### 【0080】

なお、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

#### 【0081】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

#### 【0082】

プログラムコードを供給するための記憶媒体としては、例えば、フレキシブル

ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

#### 【0083】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0084】

さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

#### 【0085】

##### 【発明の効果】

以上説明したように、本発明によれば、仮想空間の構造に基づいた排他的制御権の取得方法を提供することにより、操作する側に仮想空間の構造を特別に意識させることなく、必要な排他的制御権を適切に取得することが可能となる。

##### 【図面の簡単な説明】

##### 【図1】

第1実施形態におけるデータベース共有システム全体の構成を示すブロック図である。

##### 【図2】

第1実施形態によるプロセス間の基本的な情報伝達方法を示す図である。

##### 【図3】

第1実施形態による端末装置のハードウェア構成を示すブロック図である。

**【図 4】**

第 1 実施形態によるデータベースの更新手順を示す図である。

**【図 5】**

イベントデータのデータ構成例を示す図である。

**【図 6】**

操作キューのデータ構成例を示す図である。

**【図 7】**

第 1 実施形態によるクライアントプロセスのフローチャートである。

**【図 8】**

図 7 のクライアントプロセスにおける操作処理を示すフローチャートである。

**【図 9】**

図 7 のクライアントプロセスにおける受信イベント処理を示すフローチャートである。

**【図 1 0】**

第 1 実施形態によるサーバプロセスを説明するフローチャートである。

**【図 1 1】**

データベースの記述ファイルフォーマットを示す図である。

**【図 1 2】**

仮想空間の構造の例を示す図である。

**【図 1 3】**

図 1 2 の仮想空間を表現するシーングラフの例を示す図である。

**【図 1 4】**

操作対象別に排他的制御権を設定する例を示す図である。

**【図 1 5】**

第 1 実施形態による排他的制御権を取得する手順を説明するフローチャートである。

**【図 1 6】**

排他的制御権を解放する手順を説明するフローチャートである。

**【図 1 7】**



サブシーングラフの排他的制御権を取得する例を示す図である。

【図 1 8】

サブシーングラフの排他的制御権を取得できない例を示す図である。

【図 1 9】

サブシーングラフにおける排他的制御権の取得及び解放を説明する図である。

【図 2 0】

サブシーングラフにおける排他的制御権の取得及び解放を説明する図である。

【図 2 1】

サブシーングラフにおける排他的制御権の取得及び解放を説明する図である。

【図 2 2】

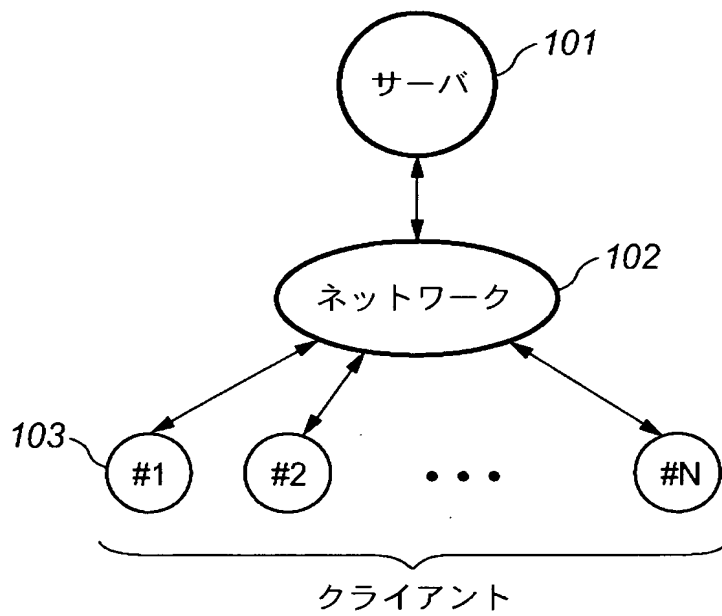
第 2 実施形態による排他的制御権を取得する手順を説明するフローチャートである。

【図 2 3】

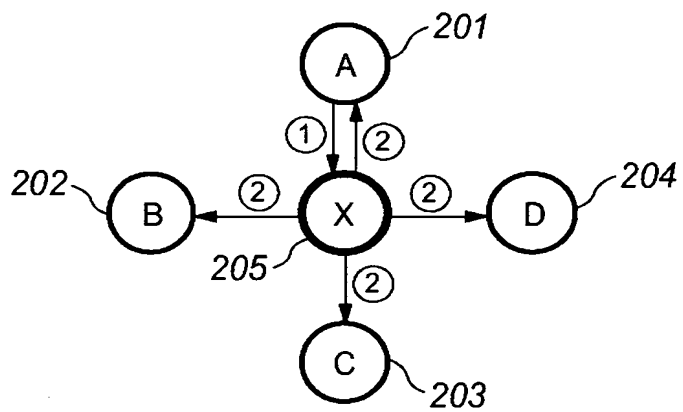
図 1 5 に示した処理において、再帰呼び出しが必要となるシーングラフの一例を示す図である。

【書類名】 図面

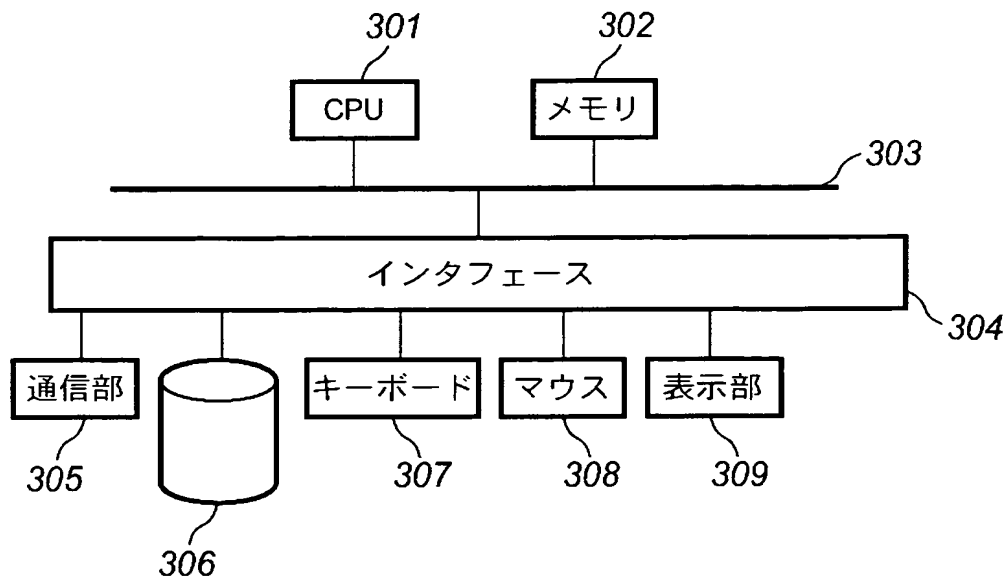
【図 1】



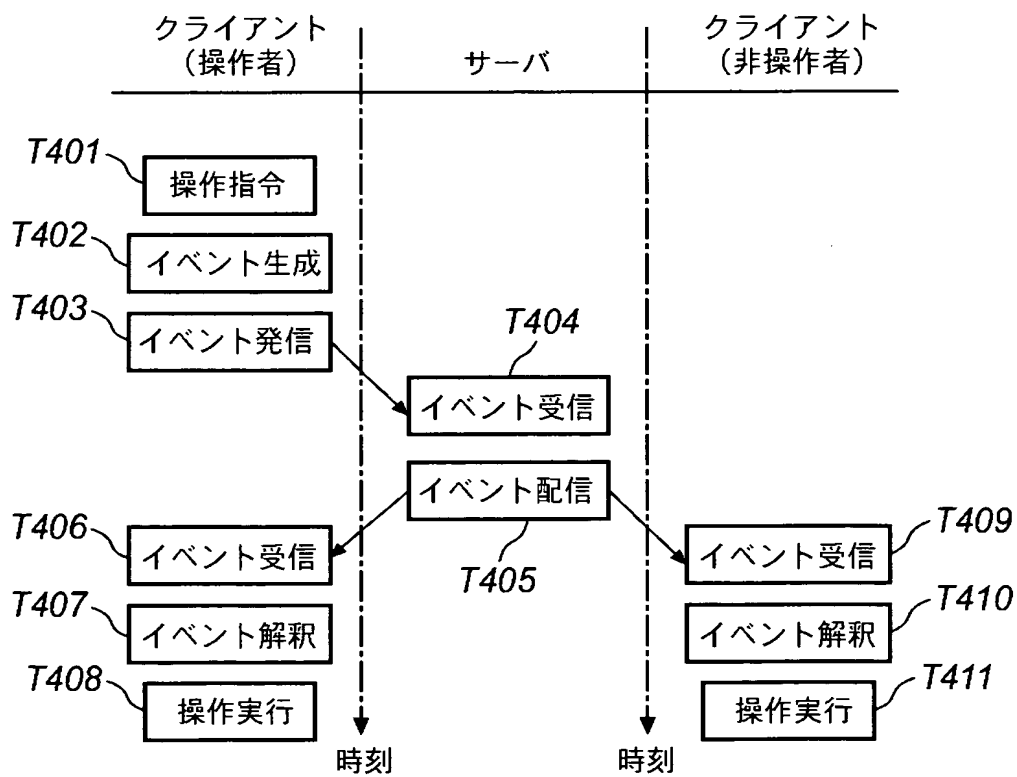
【図 2】



【図 3】



【図 4】



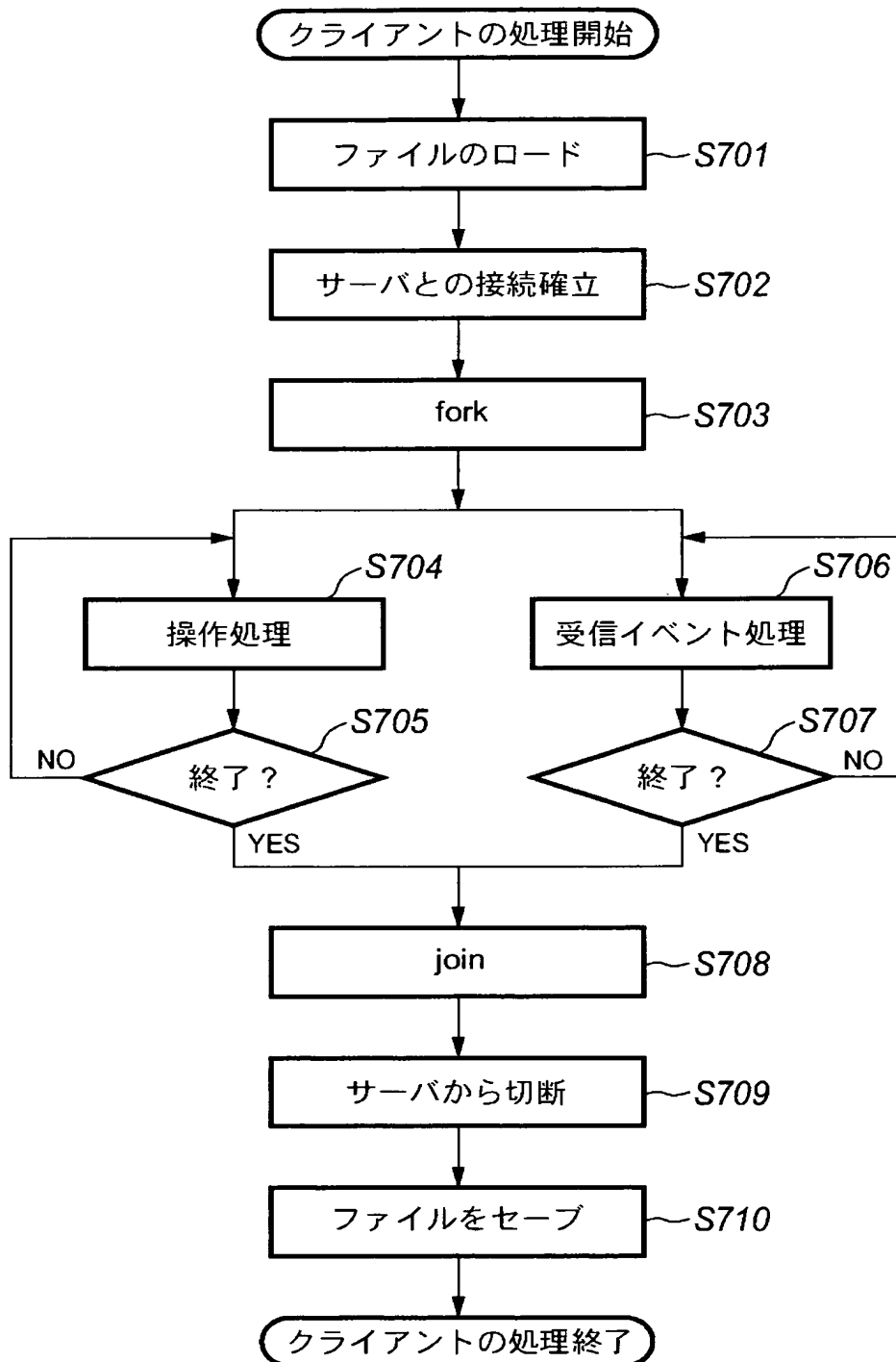
【図 5】

クライアント ID	〜 501
操作 ID	〜 502
エントリ ID	〜 503
操作内容	〜 504

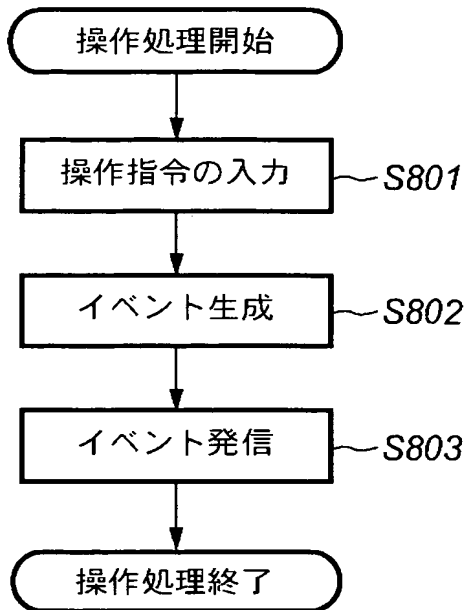
【図 6】

キューアイテム No.1	キューアイテム No.2	....	キューアイテム No.m
-----------------	-----------------	------	-----------------

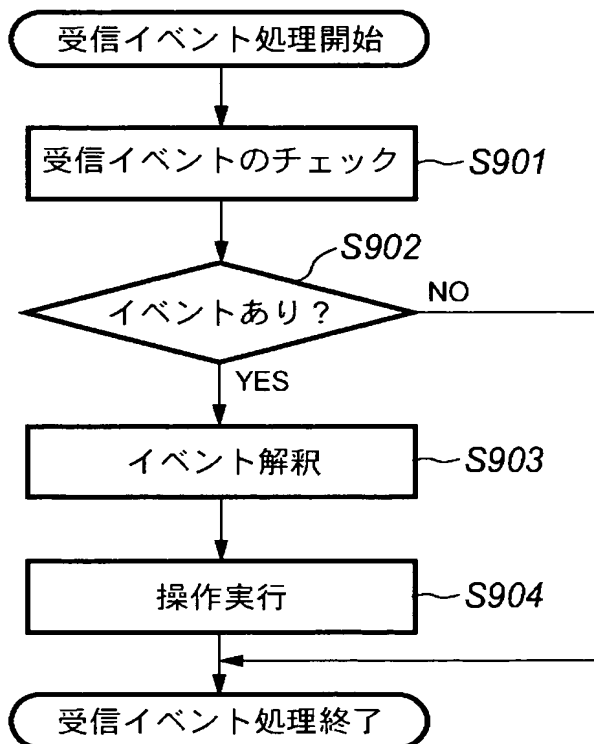
【図 7】



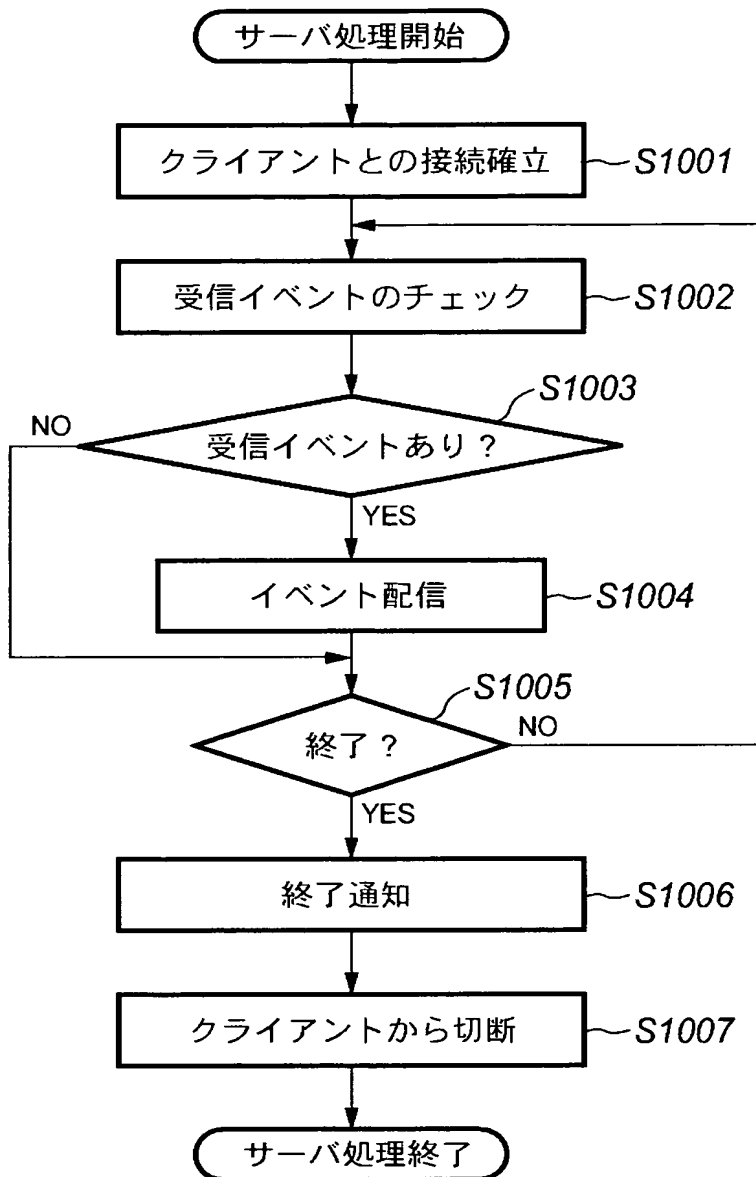
【図 8】



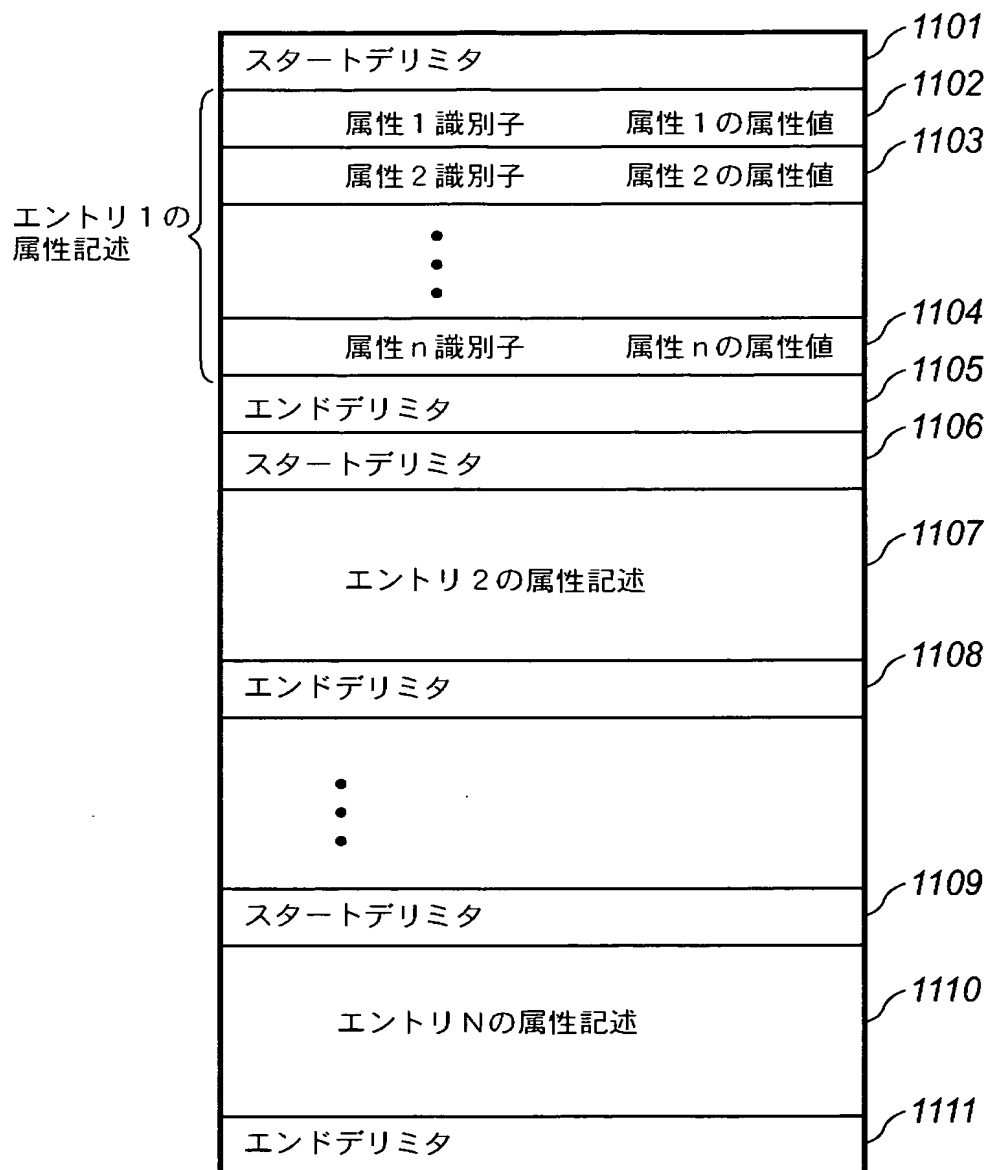
【図 9】



【図 10】

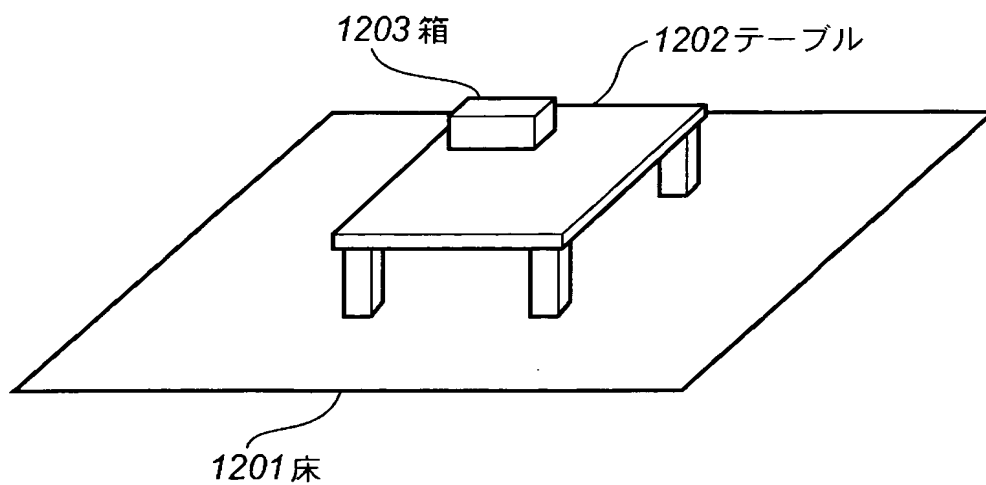


【図 11】

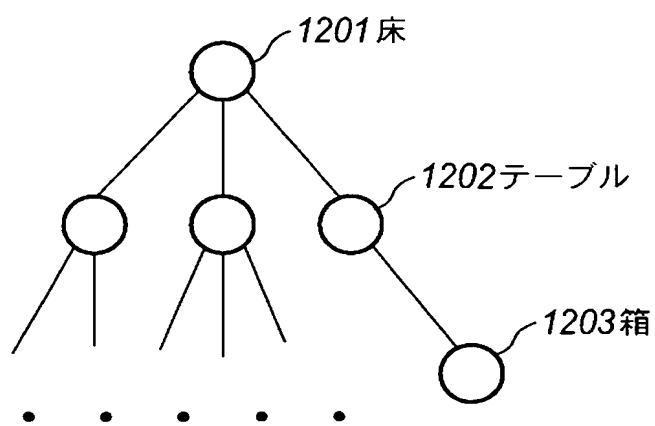




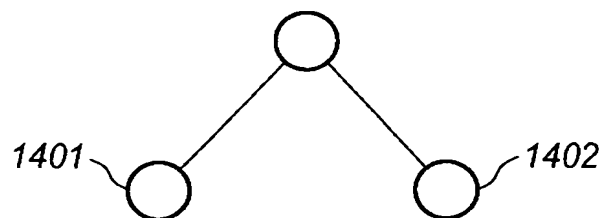
【図 1 2】



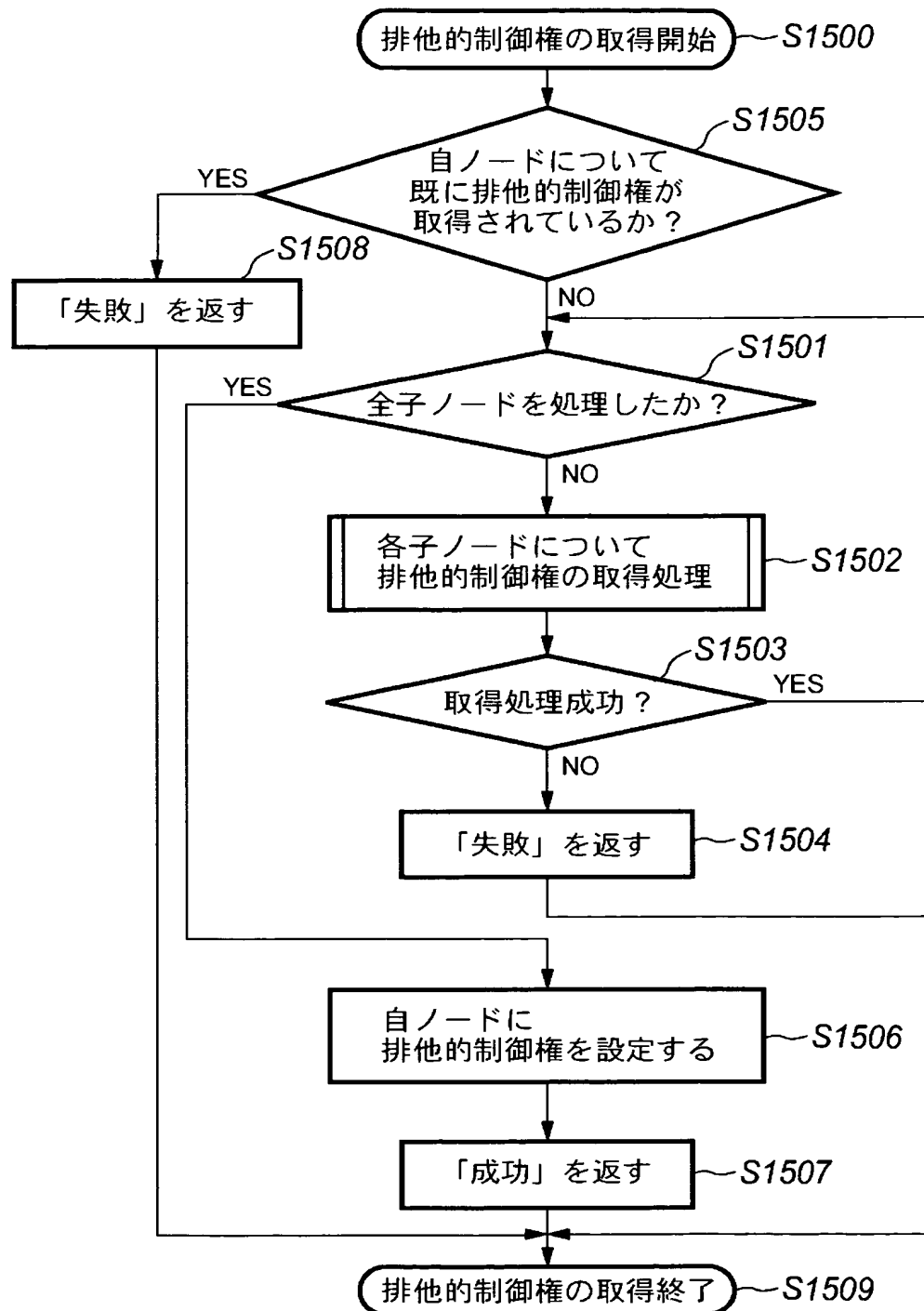
【図 1 3】



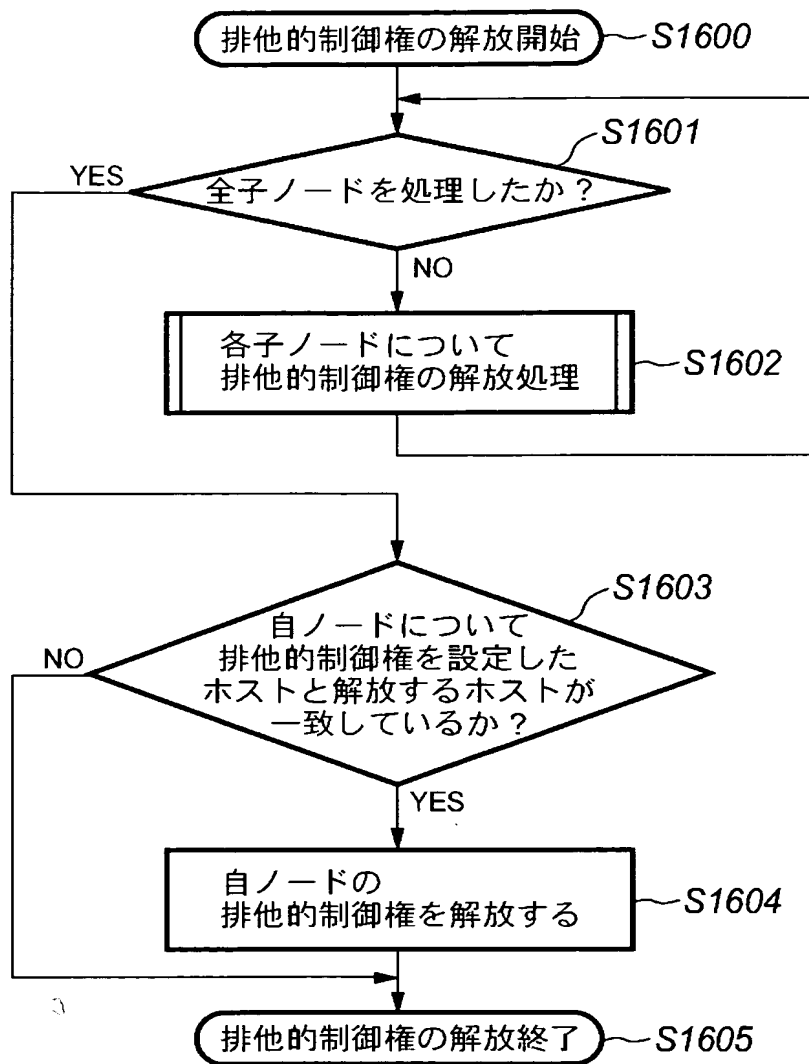
【図 1 4】



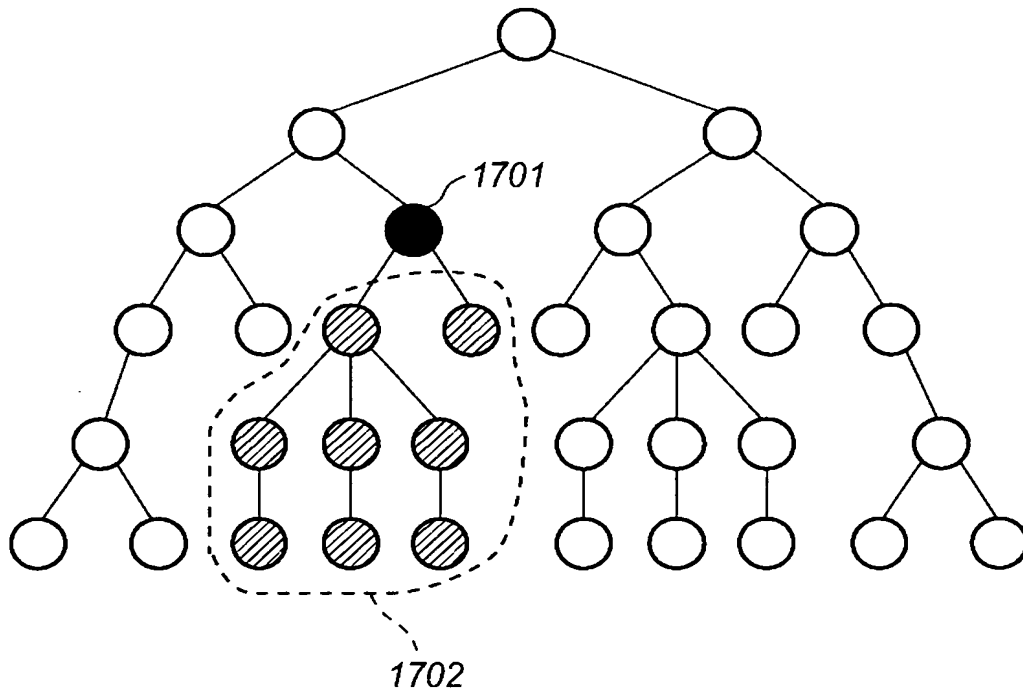
【図 15】



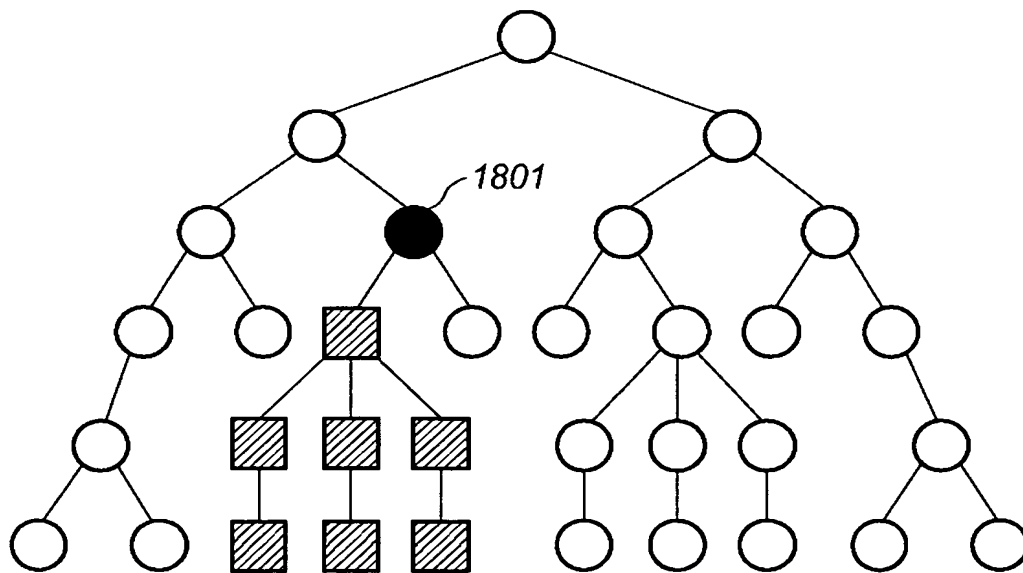
【図 16】



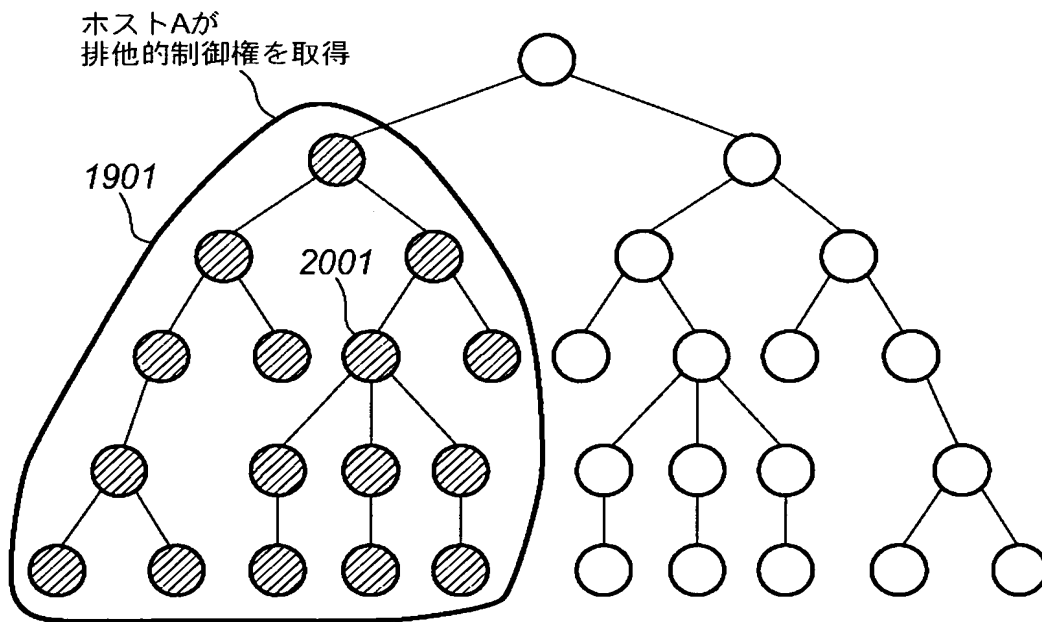
【図 17】



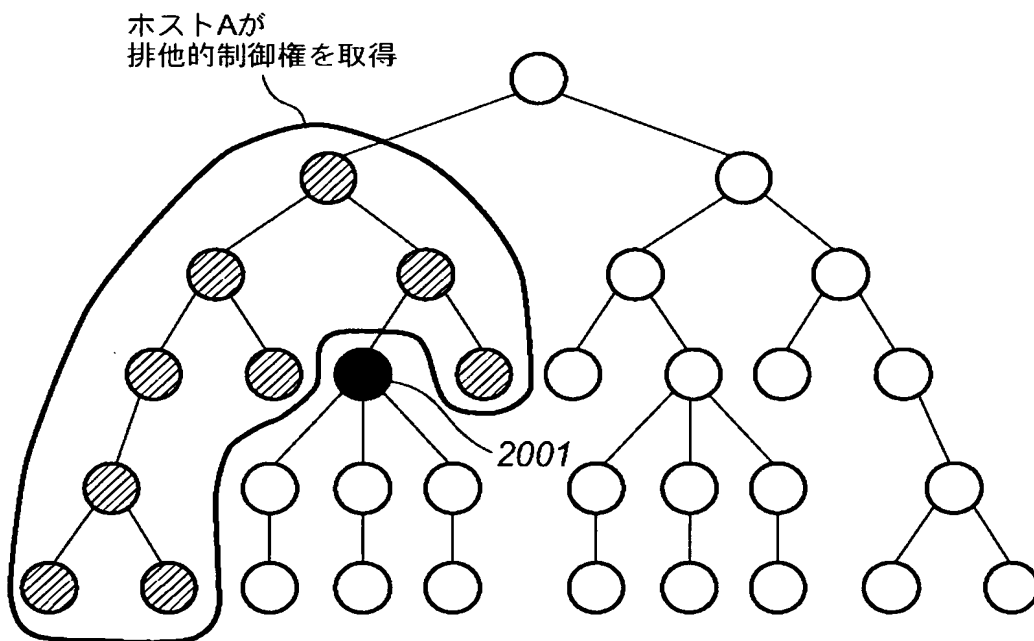
【図 18】



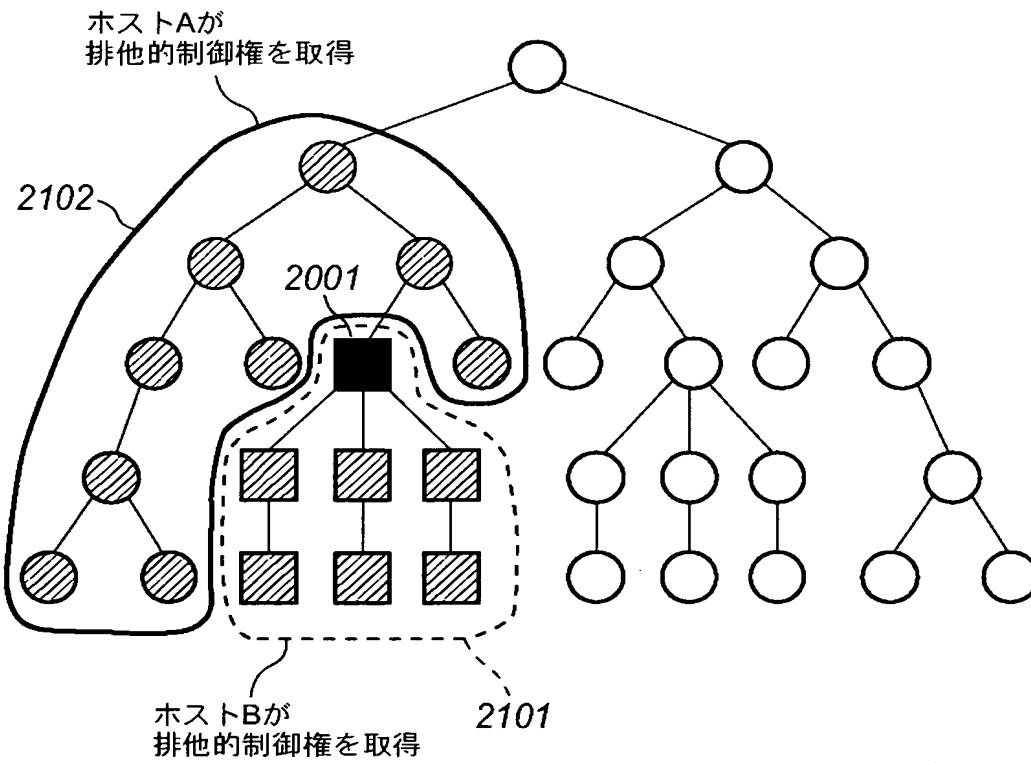
【図 19】



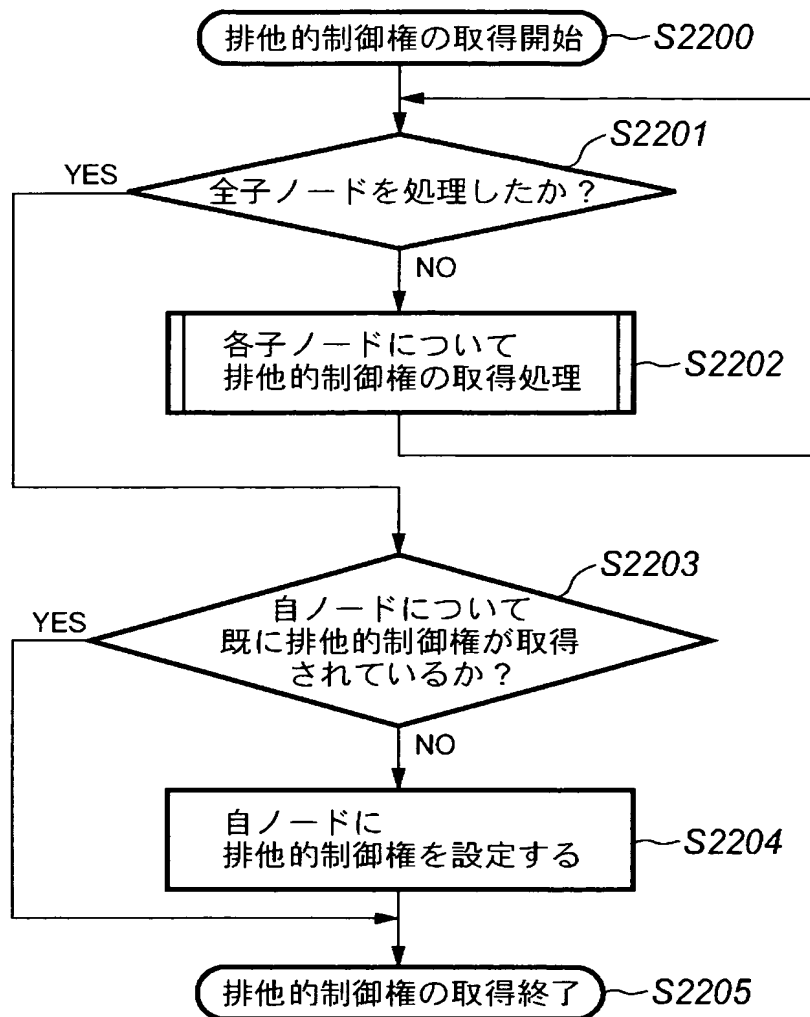
【図 20】



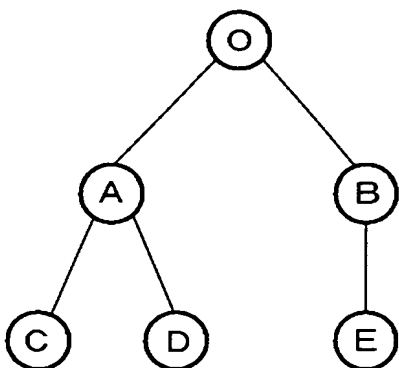
【図 2 1】



【図 2 2】



【図 2 3】



【書類名】 要約書

【要約】

【課題】 仮想空間の構造に基づいた排他的制御権の取得方法を提供することにより、操作する側に仮想空間の構造を特別に意識させることなく、必要な排他的制御権を適切に取得することを可能とする。

【解決手段】 ネットワークを介して接続された複数のクライアントによって複数のデータアイテムで構成されたデータを共有するシステムにおいて、特定プロセスによるデータアイテムの排他的制御権を設定するための情報処理方法が提供される。この方法において、排他的制御権を設定するべく所望のデータアイテム（1701）が指示されると、複数のデータアイテムの階層構造情報に基づいて、指示されたデータアイテムよりも下位の階層に存在し、該指示されたデータアイテムに関連するデータアイテムが探索される。そして、指示されたデータアイテムと探索されたデータアイテム（1702）に対して排他的制御権が設定される。

【選択図】 図17



特願 2 0 0 3 - 0 5 4 1 4 0

出 願 人 履 歴 情 報

識別番号

[ 0 0 0 0 0 1 0 0 7 ]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社